Towards Data-Driven Generation of Visualizations for Automatically Generated News Articles

Alhalaseh, Rola

ACM, Association for Computing Machinery
2018-10

Alhalaseh , R , Munezero , M D , Leinonen , M , Leppänen , L J , Avikainen , J E O & Toivonen , H 2018 , Towards Data-Driven Generation of Visualizations for Automatically Generated News Articles . in Proceedings of the 22nd International Academic Mindtrek Conference . ACM, Association for Computing Machinery , pp. 100-109 , The 22nd International Academic Mindtrek Conference , Tampere , Finland , 10/10/2018 . https://doi.org/10.1145/3275116.327

http://hdl.handle.net/10138/298354
https://doi.org/10.1145/3275116.3275131

unspecified
acceptedVersion

Towards Data-Driven Generation of Visualizations for Automatically Generated News Articles

Full Paper

Rola Alhalaseh
Department of Computer Science, University of Helsinki
Helsinki, Finland
rola.alhalaseh@helsinki.fi

Myriam Munezero
Department of Computer Science, University of Helsinki
Helsinki, Finland
myriam.munezero@helsinki.fi

Miika Leinonen
Department of Computer Science, University of Helsinki
Helsinki, Finland
miika.leinonen@helsinki.fi

Leo Leppänen
Department of Computer Science, University of Helsinki
Helsinki, Finland
leo.leppanen@helsinki.fi

Jari Avikainen
Department of Computer Science, University of Helsinki
Helsinki, Finland
jari.avikainen@helsinki.fi

Hannu Toivonen
Department of Computer Science, University of Helsinki
Helsinki, Finland
hannu.toivonen@helsinki.fi

ABSTRACT

A feature news story is often accompanied by illustrations and visuals. These visualizations can be, e.g., timelines, line charts, pie charts, or images. In this article, we present a largely data-driven and domain-independent approach for generating visualizations to accompany automatically generated news articles. We demonstrate the feasibility of our approach by applying it to statistical data on crime in Finland. The practical implementation demonstrates how the automatically generated visualizations provide additional information and interactivity to the news articles. We further illustrate how the approach presented is easily transferable to different domains with structured numerical datasets.

CCS CONCEPTS

• **Human-centered computing** → **Visualization**; • **Applied computing** → **Media arts**;

KEYWORDS

Visualization, automation, locator map, graph, visualization in journalism

ACM Reference Format:
Rola Alhalaseh, Myriam Munezero, Miika Leinonen, Leo Leppänen, Jari Avikainen, and Hannu Toivonen. 2018. Towards Data-Driven Generation of Visualizations for Automatically Generated News Articles: Full Paper. In *Academic Mindtrek 2018 (Mindtrek 2018), October 10–11, 2018, Tampere, Finland*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3275116.3275131

Mindtrek 2018, October 10–11, 2018, Tampere, Finland

ACM ISBN 978-1-4503-6589-5/18/10…$15.00
https://doi.org/10.1145/3275116.3275131

1 INTRODUCTION

In this work, we investigate the challenge of automatic generation of visuals to accompany textual news articles. These news articles are themselves produced automatically using a natural language generation (NLG) system [13] that takes as its input structured, statistical data and outputs textual stories that describe the most newsworthy aspects of the dataset.

It is common to have visuals accompany the text content of human-produced news articles. These visuals either summarize parts of the story or provide additional information to the news article [7]. For example, the inclusion of maps can give insight and context on the geographic location of the story, especially when the location is not familiar to the reader [8].

Visualizations serve as a tool to help with understanding and communicating data to others [3]. They can assist when dealing with large amounts of data [11] by making it easier for the reader to observe trends, patterns or outliers in the data [2], since the human mind is able to create a mental image and absorb patterns and information quickly through these illustrations [21]. Describing these same trends, patterns or outliers verbally could easily result in text that is too long and hard to follow. Furthermore, illustrations allow for easy comparisons between different attributes in the presented data [15, 17], something that is increasingly difficult to accomplish with text as the number of compared points of data increases.

Automatically producing good visuals that fit the purpose and convey the information in an understandable way is a challenge. In particular, a challenge remains in choosing *what* to visualize. Our contribution in this paper is to present a data-driven approach to generating visualizations that accompany and enhance textual content. This contribution is in two parts. First, we describe an automated method to generate locator maps that show the geographic focus of an article, together with an automatically determined point of reference, to the user. This serves to increase the user's geographical awareness for the location of the news article. Second, we describe an automated approach to the generation of line and bar charts from time series data. Our approach is based on automated detection of newsworthiness and relevancy in the given

data and use of interactive elements to allow the reader to gain more information by interacting with the visualized data points. We exemplify both the locator maps and the graphs using an implementation in the domain of crime statistics, providing a description of the implementation details and examples of output.

In the next section, we give an overview of related previous work. The contribution of this paper is then presented in two parts: First, Section 3 describes the proposed data-driven approach for visualizing the data using both locator maps and graphs. Next, Section 4 describes a detailed implementation of the approach in the context of crime statistics. Finally, Section 5 discusses these methods and our observations in a broader setting and suggests future enhancements.

2 PREVIOUS WORK

Visualization is a term referring to the process of representing a situation or any kind of object as well as a set of information in a graphical and visual form [6]. In the context of this work, visualization involves providing context to a textual news article, in the form of a locator map, and by representing complex data in a graphical format.

While both automatic news generation and automatic visualization have been studied in the past, we are not aware of works that produce visualizations in a data-driven approach. However, previous works on the generation of locator maps and other visualizations in general do exist. The following sections provide an overview of these previous works.

2.1 Generation of Locator Maps

According to Arthur Robinson, '[a]ll locations are relative, and therefore they must be established in relation to some reference[..]. If such a point is determined every other point on the surface can be located in terms of direction and distance from this point' [1]. Locator maps are maps that allow a user to locate a point (e.g. a municipality) previously unknown to them by displaying it on a map that also contains a point that is already known to them (e.g. a nearby large city). In the context of news, such maps have previously been shown to improve the readers' awareness of geographical context, as they allow the readers to easily understand the location referred to by the article [8].

The automated generation of locator maps has been studied previously in the context of human-written articles by Gao et al. [7], where they describe an automated pipeline that generates interactive and annotated maps given context articles. They further present an implementation of the pipeline, NewsViews, where they mine the location and the topic of interest from a news article, and the system then generates relevant visualizations to accompany the article.

Picozzi et al. [16] present a case study that utilized traffic data to generate an interactive and data-driven web mashup visualization of the spatiotemporal data using a combination of maps, graphs and a calendar. The user has the ability to select a certain place on the map, time or time period.

2.2 Generation of Graphs and Charts

Several approaches for automatic generation of visualizations have been proposed in the literature. One of the earlier works is the SAGE program by Roth et al. [20]. SAGE is a knowledge-based graphical display system that combines diverse information (e.g. quantitative, relational, temporal, and geographic) to automatically generate graphs from data. However, they consider their approach to be complex for users with limited knowledge in graphs since it would be much easier for an expert to use the system and customize the graphs generated according to their needs. In our system, the users are not required to have experience with determining which graph is suitable in order to be able to obtain data and further details from the graphs.

Other works in the literature include that of Sun et al. [22], who propose a semi-automated visual analytic model in which they apply a heuristic graph generation algorithm to create a visualization based on a user query, and the Contextifier system by Hullman et al. [10] which produces visualizations of stock behavior based on textual content.

Furthermore, some systems for automated news production also automatically produce simple visualizations. For instance, the ice hockey news generator[1] of the Finnish Broadcasting Company YLE automatically generates an image depicting the final score of the match to accompany the textual articles.

Few of the approaches utilized in the previous work employ a data-driven approach in deciding what to visualize. In this paper, we propose such an approach and apply it to generate different types of visualizations, in order to illustrate its potential. While our demonstration is in the crime statistics domain, the design of the data-driven approach is easily applicable to other structured datasets from other domains.

3 DATA-DRIVEN ARCHITECTURE

In this section, we describe the overall data-driven approach developed for the automatic generation of visualizations; details regarding the implementation of this architecture are given in the following section.

The visualization generator is developed as a complementary component to an NLG system that automatically generates news articles from structured data based on a user query. This system's architecture is based on the architecture presented by Leppänen et al. [13], with added visualization components. Changes were made to the natural language generation components as well, but these are not relevant in terms of the contribution of the present paper. The extended system is presented in Figure 1. The architecture follows a pipeline structure where raw data and relevant parameters are fed in at the start. Then, a series of transformations to the data is performed, resulting in a news article.

As shown in Figure 1, the visualization generation component consists of two sub-modules: the *graph generator*, which is domain- and language-independent; and the *caption generator* which is domain-independent but language-specific. Note that while the caption itself is obviously domain-specific, the generator for captions is not: all the domain-specific information is given as parameters in the form of a domain-specific lexicon.

[1] https://github.com/Yleisradio/avoin-voitto

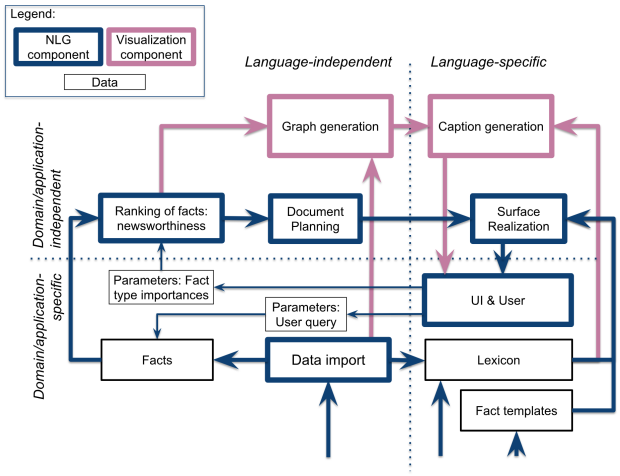

Figure 1: Overview of the architecture. Thick boxes represent software components and thin boxes data structures. Adapted from [13].

The system processes data in atomic elements called *facts*, which contain a value and associated metadata on what the value describes. More specifically, a fact captures the *who*, *what*, *where*, and *when* of a piece of information. For example, given crime statistics as the input data, a single fact could capture information that a total number of offenses of a certain crime happened in a specific municipality during some specific time. In a sense, these facts are the minimal independently meaningful pieces of information in the architecture.

The graph generation component receives as input the most newsworthy fact from the fact-ranking component. The ranking component is responsible for estimating the newsworthiness of all of the facts relevant to the user query and is shared with the part of the pipeline that produces the textual content. In the context of crime statistics (detailed more in Section 4), a query typically identifies a municipality, and the task of the fact-ranking component is then to identify the most newsworthy aspect of the crime data in that municipality.

Using the method described in Leppänen et al. [14], newsworthiness is determined as the product of four factors: *topicality, outlierness, interestingness, and personalization.* Topicality details how important a fact is in terms of the current (public) discourse or time, while outlierness captures how unexpected, surprising or rare the fact is. Interestingness captures the intrinsic (as estimated by e.g. a journalist) newsworthiness of the fact and personalization mirrors the aforementioned interestingness but reflects the preferences of the individual reader.

Additionally, the graph generator has access to all of the imported data in order to utilize all of the data points related to the selected fact. For instance, based on the *where* information in a fact, the graph generator can create a locator map to show the location of the event. With the *what* and *when* information, the graph generator can generate various graphs such as line graphs, bar charts, pie charts, etc., to show and compare events in different locations at different times. Deciding which graph type to use depends on what best represents the imported dataset and the message to be communicated.

The caption generation utilizes the *what* and *when* information of the selected fact together with a lexicon to produce the linguistic expressions, such as the titles and legends, of the visualizations in the target language. Since these natural language expressions are relatively simple, at least in comparison to the textual body of the news article, this component can be left relatively simple and does not necessarily need to employ the full capabilities of all the language generation parts of the NLG pipeline.

4 IMPLEMENTATION

In this section, we provide a description of our implementation of the approach explained in Section 3 for use with statistical crime data.

4.1 Domain: Crime Statistics

In Finland, monthly data on crime statistics [4] is collected, maintained and made publicly available by Statistics Finland[2], the national statistical institution. The crime statistics data represent offenses that the police, customs, and border control have become aware of due to either someone reporting the crime or the authorities discovering the crime themselves. Thus, it does not include crimes and offenses that are not recorded by the authorities, i.e. hidden crimes.

The crime statistics are available as aggregate for the whole country and for each of the 311 municipalities in Finland. The crime statistics divides crimes into 147 groups. This number includes subcategories that indicate varying severity classes of the same crime. These were, however, ignored for simplicity of reporting and visualization. In total, 99 different crime categories remained, which we then grouped into 10 non-overlapping high-level crime categories based on the categorization used by Statistics Finland. Table 1 displays these high-level categories, their sizes in terms of different crime types included, and examples of the crimes that fall within them.

At the time of conducting this work, the latest crime statistics dataset spanned from January 2009 to December 2017. We also obtained the per-municipality population data starting from 2010 in order to calculate statistics such as per capita crime rates.

4.2 Data Preparation and Selection

The system transforms each data point from the crime statistics to the generic fact structure that captures the *who*, *where*, *what*, and *when* aspects of a piece of information.

The values are divided into two distinct categories: the main values and the *type* values. The main values contain the actual data, and the accompanying *type* values contain information on how to interpret the data. Within the domain of crime statistics in Finland, the location has two possible types (`where_type`): country and municipality, with the value (`where`) holding the name of the location (e.g. 'Finland' or 'Helsinki'). Similarly, time can have two different types (`when_type`), a year or a year/month pair, with the value (`when`) containing the actual time value, e.g. '2015' for the year 2015 or '2015M04' for the April of 2015.

[2] http://tilastokeskus.fi/index_en.html

Table 1: Crime categories with the number of crimes they contain and examples thereof.

Category	Size	Example Crimes
Crimes against life	5	manslaughter, murder
Sexual crime	3	sexual abuse of a child, rape
Bodily injury	7	assault, negligent injury
Narcotics	5	abuse of narcotics
Crimes against property	34	theft, robbery
Crimes against authority	6	perjury, resisting an official
Crimes involving alcohol	4	minor alcohol offense
Traffic-related crimes	9	drinking and driving
Other crimes	18	environmental offense
Investigations	8	inquest, missing person

The `what_types` are more varied and represent the 99 crime types and the 10 high-level categories described in Table 1. In the simplest case, the corresponding value (`what`) is simply the number of times the crime in question was committed in the location and the time period defined by the `when` and `where` fields. Using these base statistics for each crime, we also derive other `what` and `what_type` values as described below. To make the below derivations easier to explain, we shall use the crime 'murder' as a running example (i.e., `what_type` = murder).

4.2.1 Normalized Values. The data obtained from Statistics Finland gives us the crimes and the total number of times the crime has been committed during each calendar month. From these, we first calculate yearly totals and store them. Using the population data, we then normalize the values into crimes per 1,000 inhabitants for each of the municipalities for meaningful comparisons between municipalities of different sizes. This results in a new `what_type` (e.g., `murder_normalized`). Unlike the crime data, the population data is only updated quarterly, but we believe this to be of no practical consequence for our purposes.

4.2.2 Change Values. In addition to the monthly and yearly totals, we calculate changes between consecutive years at one- and two-year intervals, for example changes from 2016 to 2017, and from 2015 to 2017. The change values are calculated for normalized and unnormalized totals, both as absolute and relative change. The absolute change is the difference in the value between the two time points, and the relative change is the absolute change divided by the starting value, i.e. the change in percentages. This results in four different change statistics, as summarized in Table 2. While the two relative changes tend to be practically identical, due to the automation of this process, it is simpler to compute them than to add in logic to ignore one.

4.2.3 Value Ranks. To enable meaningful comparisons across different municipalities, crime types, and months during a year, we also calculate relative ranks. By fixing two of these three variables (location, crime, and time), a rank is calculated based on the third.

Table 2: Absolute and relative change value types generated from the normalized and unnormalized value types.

	Unnormalized	Normalized
Absolute	`murder_change`	`murder_normalized_change`
Relative	`murder_percentage_change`	`murder_normalized_percentage_change`

For example, if we fix the crime type to murder, and the time to year 2014, we can calculate which of the municipalities had the highest murder rate (per 1,000 inhabitants) during 2014. For monthly data, we can also partially limit the time to a specific year in order to rank different months during the specified year.

The ranks are calculated using the population-normalized values and both the absolute and relative change values. For totals, we calculate the ranks both in increasing and decreasing order. This allows us to easily establish the highest and lowest crime rates. For example, in Helsinki, the murder rate was 20th in rank and 8th from the bottom in comparison to other municipalities in the year 2014 (see Table 3). The low numbers in both ranks despite of the large total number of municipalities is caused by the way ranks are calculated. When two municipalities are tied, they share the same rank, and the next municipality is assigned the next rank. Therefore, all municipalities with zero murders have rank 1 in the reverse ranking, and the municipality with the smallest non-zero normalized murder count gets assigned rank 2.

For changes, we calculate separate rankings for positive (the nth largest growth) and negative change values (the nth largest decrease). As an example, Table 3 shows the change rank values between 2012 and 2014 in Helsinki, where the total number of murders in Helsinki in 2014 was seven, a drop of three from the ten murders in 2012. The drop in the absolute number of murders was the 17th largest during that time, but in a relative scale, the change was 5th largest. Note that the `increase_rank` values are not meaningful since there was no increase in the observed time period.

4.2.4 Assigning Newsworthiness. As mentioned in Section 3, we make use of the newsworthiness formulation given in [14] to assign a score to each fact. Determination of what is newsworthy is based on the product of topicality, outlierness, interestingness, and personalization factors.

Topicality is approximated by looking at the temporal elements of each fact and assigning a weight so that facts pertaining to more recent time periods are seen as more newsworthy. The topicality of a year is calculated by

$$t = \min\left(1, \frac{1}{((current\_year + 1) - year)^2}\right), \quad (1)$$

with the values for months assigned linearly between the yearly values.

Outlierness is determined using the Inter Quartile Range method, as described in [14]. This approach considers how much of an outlier

Table 3: An example showing the generated rank, change and change rank types for Helsinki. The values for rank and rank_reverse are during 2014 while the remaining change-related values are for between 2012 and 2014.

What type	Value
murder_total	7
murder_normalized	0.0113
murder_normalized_rank	20
murder_normalized_rank_reverse	8
murder_total_change	-3
murder_normalized_change	-0.0053
murder_normalized_change_increase_rank	n/a
murder_normalized_change_decrease_rank	17
murder_norm..._percentage_change_incr..._rank	n/a
murder_norm..._percentage_change_decr..._rank	5

each individual value is in comparison to a set of comparable values, with more outlying facts being given a higher weight.

Interestingness approximates human tendencies such as viewing certain crimes as intrinsically more newsworthy than others. We use a weighing scheme based on the harshness of the punishment of a crime category each individual crime belongs to. Each of the ten broad categories was assigned a weight based on the maximum punishment that that category can receive based on the Finnish Penal Code [5] at the present time. Since the information is used as a proxy for the interest of a present day reader, we do not use the historical maximum punishments for historical crimes: we are interested in how the society views a certain crime now, rather than at the time the crime was committed. The categories and example crimes within them are presented in Table 1 and the weights used are presented in Table 4. We also use weights to encourage statements of the type 'Xth most' and penalize statements of the type 'Yth least'. Additional weights are used to, for example, prefer yearly data over monthly data.

For the *personalization factor*, we allow the user to explicitly define which geographic location they want a story to describe, thus effectively setting the personalization factor (and consequently, the total newsworthiness) to zero for facts pertaining to locations other than the one they selected.

Ranking each fact using a product of the aforementioned four factors, each producing a real-valued number, we select the fact with the highest ranking. This as-newsworthy-as-possible fact is used as the input for the visualization component to create a number of visualizations that are described in the next sections.

4.3 Locator Maps

Since crime statistical data is available for each of the 311 municipalities in Finland, a large amount of the locations are likely to be such that the readers either have only a vague idea of their location or do not know it at all. Thus, we decided to include a locator map

Table 4: Crime types and their pre-assigned interestingness weights based on maximum punishment.

Crime Category	Max. punishment	Weight
Crimes against life	Life imprisonment	5
Sexual crimes	10 years	4
Bodily injury	10 years	4
Narcotics	10 years	4
Crimes against property	4 years	3
Crimes against authority	4 years	3
Crimes involving alcohol	4 years	3
Traffic-related crimes	2 years	2
Other crimes	4 years	3
Investigations	n/a	n/a

to accompany each news article so as to familiarize the reader with the geographical location of any municipality.

The locator map component produces a map displaying the location referenced by the news article together with a reference location the user is assumed to know. The map is zoomed so that it only fits those two locations and a small margin. Figure 2 shows an example of such a locator map, where the small municipality of 'Honkajoki' is the location discussed in the article and 'Pori' as the reference landmark. For articles about one of the reference locations, the nearest reference location would be the location itself and thus the zoom level would be too high for a meaningful map. Thus, for these reference locations, the generator instead displays a map of the whole country with a pin designating the location discussed in the story.

For the implementation of the locator maps, two processes were involved: 1) The creation of a list of reference locations, and 2) the creation of the locator map based on the user query.

4.3.1 Creating List of Reference Locations. The selection of the reference location is of utmost importance in the construction of the locator map: if the reader is unfamiliar with the selected reference location, the map fails to convey the necessary geographical information. We elected to create the list of reference locations automatically, using the population count of a municipality as a proxy for how well known a municipality is. Potential ways to improve this method will be discussed in Section 5.

To generate the list of reference locations, we use a simple algorithm that uses three parametres to adjust the selection of reference points: d_{min} (the minimum allowed distance between reference locations), d_{max} (the maximum distance from any municipality to the closest reference location) and n_{max} (the maximum number of selected reference locations). The d_{max} and n_{max} parameters are optional. First, the algorithm sorts the municipalities in descending order by population. Then the algorithm chooses the first (i.e. largest) city in the list as a reference location. It then adds it to the (initially empty) list of selected reference locations and removes it from the list of municipalities together with all municipalities

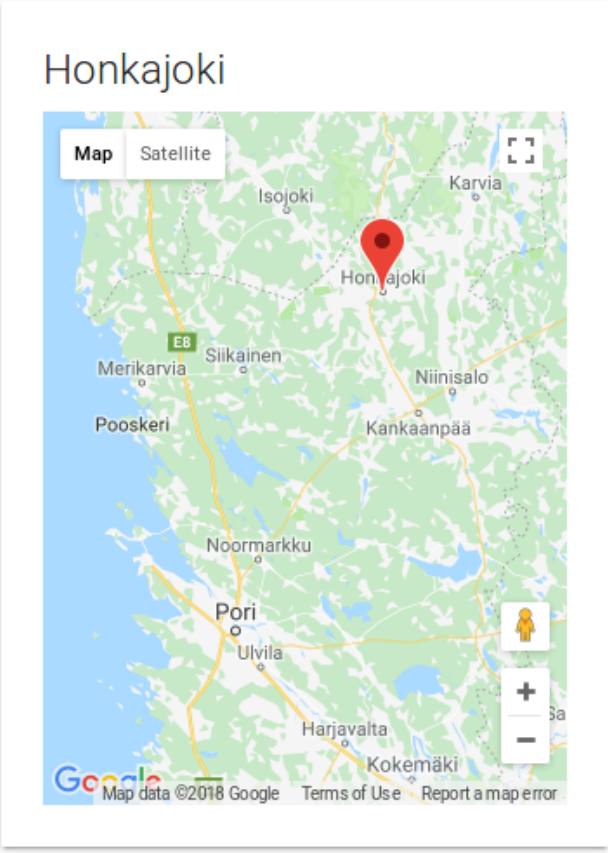

Figure 2: A locator map showing the location of Honkajoki (designated by the pin) in relation to Pori, which is the nearest reference location.

that are closer than d_{min} to the chosen reference location. This prevents the reference locations from forming tight clusters in the most densely populated areas and ensures a better coverage of the whole country. We elected to use d_{min} = 100km based on empirical trials. Smaller tested values, such as d_{min} = 50km, caused unnecessary clustering of reference location to the more densely populated southern parts of Finland. Larger distances on the other hand resulted in reference locations that were both too far apart to feel natural and in extreme cases only in a few reference points, since significant amounts of the country get immediately discarded after the first point is picked.

After each addition to the list of reference locations, the algorithm checks for possible stop conditions using the other two parameters. If all remaining municipalities have a reference location closer than d_{max}, or n_{max} reference locations have already been selected, the algorithm is finished. If neither of the conditions have been reached, the algorithm continues by selecting new reference locations from the remaining candidates and recalculating the distances until there are no candidates left or one of the stop conditions is reached. For this work, we limited the number of landmarks to 15 and set $d_{max} = d_{min}$, but the optimal settings remain an open problem.

4.3.2 Creation of Map Based on User Query. When producing a locator map, we take the user's initial query (i.e., the location for the article) and obtain its latitude and longitude coordinates. Then we find the closest reference location using the haversine formula [19]. The distance d between two points is calculated as

$$d = 2r \arcsin \sqrt{\sin^2\left(\frac{\Delta\phi}{2}\right) + \cos\phi_1 \cdot \cos\phi_2 \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right)} \quad (2)$$

where r is the radius of Earth, ϕ_1 and ϕ_2 are the latitudes of the two points, $\Delta\phi$ is the difference in latitude between the two points and $\Delta\lambda$ is the difference in longitude between the two points. ϕ_1, ϕ_2, $\Delta\phi$ and $\Delta\lambda$ are all measured in radians. While Equation 2 is not the exact geodesic distance, it should avoid any significant rounding errors for distances larger than a few meters.

For the drawing of the locator map, we use the Google Maps API[3] because of its ease of use, interactivity features, and detailed API documentation and code examples. The API allows us to draw a map based on the longitude and latitude coordinates of the queried location and the selected landmark, displays both locations on the map. We also indicate the location of the article with a pin (shown in Figure 2). The boundary of the map and the zooming level are set automatically such that both locations are visible on the map with a small margin.

4.4 Visualization of the Crime Statistics Data

In Section 3, we presented an approach for selecting a maximally newsworthy fact to visualize. Based on the `what_type` value of the fact, e.g., murder, the graph generator determines the broad category this crime value belongs to, if it is not already a broad category. Continuing to use murder as an example, the graph generator determines that the broad category is in this case 'Crimes against life' (see Table 1). We elected to graph only the high-level categories. The most newsworthy crimes are often so rare as to only occur once or twice in the dataset for any given municipality, and would result in graphs with very little additional information beyond that present in the accompanying text.

The generator proceeds to access all the population-normalized data points from 2010 and 2017 belonging to the given crime category. The system then produces one of two possible time series graph types with associated titles and legends. The following paragraphs describe these interactive time series graphs, the first displaying monthly data and the second yearly data.

4.4.1 Monthly Graph. Figure 3 displays a line graph showing monthly crime data from 2010 to 2017, with an additional smaller bar graph in a tooltip. Three lines are displayed in the primary graph, all normalized by population: 1) crime totals for the location selected in the user query, 2) average for the same crime category in municipalities that are similar to the selected location (see below), excluding the target municipality, and 3) average for the same crime over the whole country, again excluding the target municipality.

[3] https://cloud.google.com/maps-platform/maps/

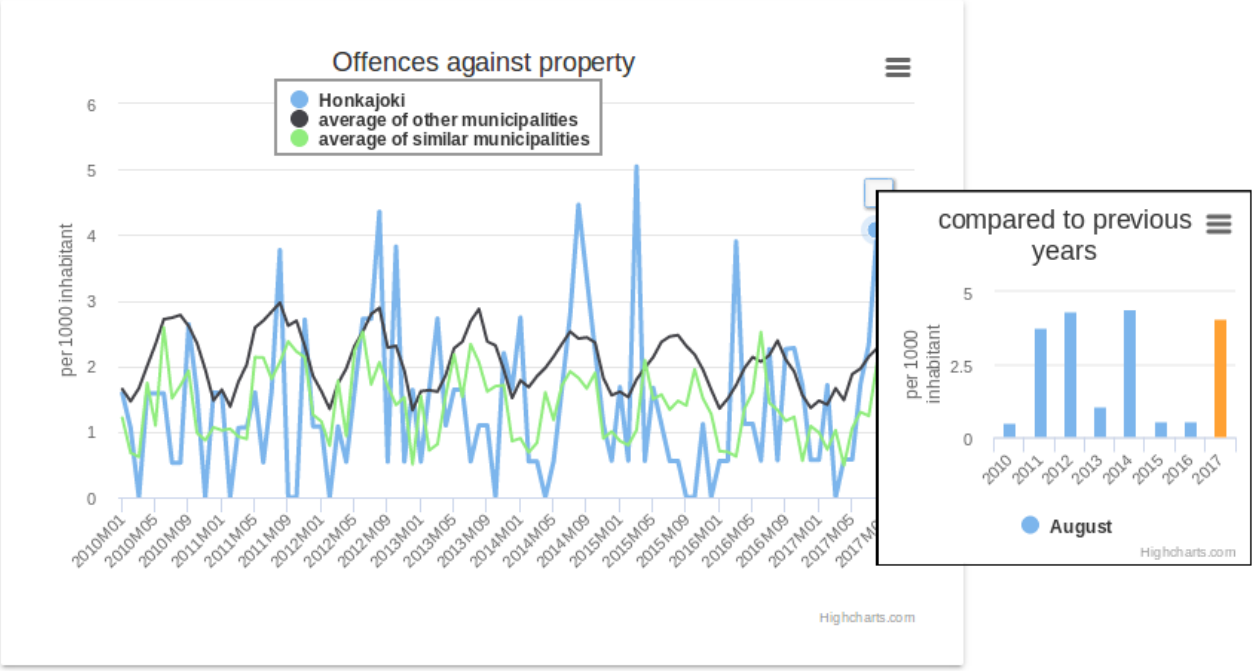

Figure 3: A graph displaying monthly time series, with a tooltip bar chart comparing the same calendar month across years (August 2017 is highlighted as the selected data point).

The system determines the group of 'similar' municipalities in a data-driven fashion using the population counts of the municipalities. Here, municipalities with populations within 10% of the target municipality are considered similar. The three comparative lines allow the reader to determine whether the target municipality behaves in an abnormal way compared to both the national average (which would naturally contain municipalities that are significantly different from the target), and to otherwise similar municipalities.

In addition, the user is able to obtain more information by hovering their cursor over any point on the time series. Doing so, they are presented with a tooltip containing another graph. This secondary graph shows a bar chart that compares the month of the selected data point to the same calendar months in different years. This allows the user to make long-term comparisons that ignore the intra-year cyclical nature of the data (i.e. certain crimes happening more commonly during certain times of the year). To improve legibility, the year for the selected data point is highlighted in a different color.

4.4.2 Yearly Graph. The second graph (Figure 4) visualizes the yearly totals of the crime category (rather than the monthly totals as in the previous graph). This gives a less cluttered overview of the change from year to year while losing the ability to observe intra-year cycles in the main graph. Similarly to the monthly graph, three time series are displayed reflecting the location the article is discussing, similar municipalities, and the whole country.

Also similar to the monthly graph, the user can select any data point in the yearly graph by hovering their cursor over it. This produces a bar chart in the tooltip graph showing the monthly values for that year, thus allowing the user to get an understanding of whether the crime in question has a seasonal component or not.

4.4.3 Generating the Visualizations. For generating the graphs in Figures 3 and 4, we use Highcharts [9], a JavaScript library that automates most of the work in generating and displaying interactive graphs.

The textual elements in the graphs, e.g. titles and legends, are generated by the Caption Generator component as described in Section 3. The component simply takes as input the relevant metadata, such as the location and the high-level category of crimes being visualized, and consults a language-specific list of template phrases ('`Crimes in year {year}`') and the lexicon of the general NLG system to obtain the terms used to describe the crimes in the graph ('Offences against property'). These are then combined to form the natural language phrases needed in the graph.

Both the numeric data and the textual elements required for the graph to be drawn are then provided as parameters for the Highcharts library, which handles the practical drawing of the graph. Overall, the generation of the whole article (both text and graphics) takes at most a few seconds, with the graphics taking at most a few hundred milliseconds of that time. Notably, this speed is achieved by the virtue of conducting a once-off preprocessing pass over the data. While this preprocessing can take up to a few hours for our data set on the scale of 5 million initial data points (expanded to multiple times that size during the preprocessing), it only needs to be done when the underlying dataset is updated, i.e. in our case every few months.

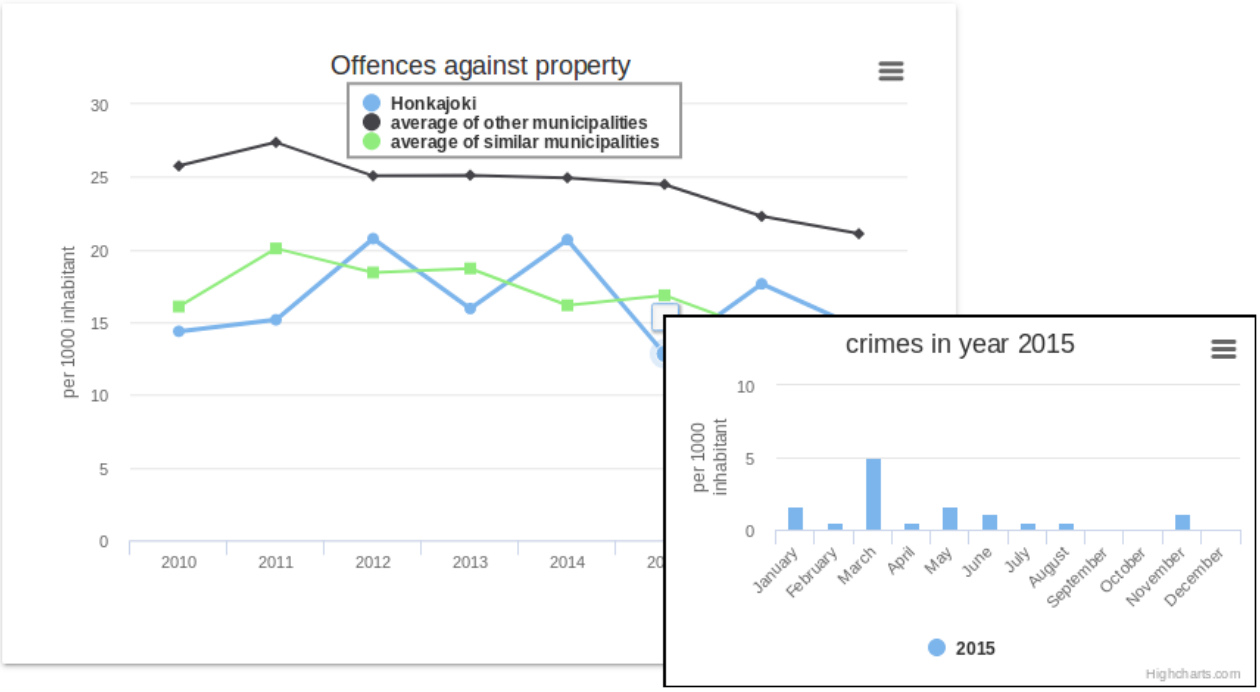

Figure 4: A graph showing yearly time series with a tooltip bar chart showing the data in each calendar month for the year of interest (2015 in the example).

5 DISCUSSION

In this paper, we presented a data-driven approach to generating visualizations from a large, temporal dataset. Further, we demonstrated an implementation of the approach to generate multiple visualizations in the context of crime statistics data for Finland.

In our view, the visualization process is composed of three steps: deciding *what* to visualize (i.e. picking the subset of data to visualize), deciding *how* to visualize, (i.e. selecting a type of chart or map that conveys the necessary information in the easiest way possible) and finally the act of creating the visualization. Formulated thus, the process bears a striking similarity to how the generation of natural language is often divided into the three tasks of content selection, document planning and surface realization [18]. While the third of these phases, the actual realization of the graph, has been for some time well handled by different visualization libraries, the first two, selecting the 'what' and the 'how', are by no means solved problems.

5.1 What to Visualize

Deciding what to visualize as a graph is a non-trivial problem. A good subject for a visualization is at the same time both interesting by itself and something that is best conveyed as a visualization. In other words, a boring visualization is a bad visualization, as are both unclear visualizations and visualizations where the main point would have been easier to convey in some other format. Both of these aspects are very human in nature, and are based on relatively soft and fuzzy values and are thus hard to determine by computation.

In this work, we have addressed the first problem of finding something interesting to visualize by employing a method designed to identify newsworthy points of data for text generation. There are both positives and negatives to this approach. As a positive, since we use the same method for determining newsworthiness for both the text generation and the visualization, the textual article and the visualization are in sync and make thematic sense. At the same time, determining newsworthiness on a per-datapoint basis forces us to consider each point of data in isolation. This may result in suboptimal results in cases where the graph presents a single interesting phenomenon, rather than multiple slightly less interesting phenomena. To avoid this, the decision on what to graph would need to be made over all the data points to be displayed, rather than only the most interesting data point.

In determining whether something is better conveyed by visualization or by text, our approach addresses the issue in an effective way. Let us assume, for example, that the newsworthiness determination process identifies as the most interesting data point the fact that some number x of murders happened in a municipality during a particular year. In this case, our visualization component does not even attempt to visualize this information alone. Rather, it assumes that since this single point of data is interesting, then consequently the whole time series to which the data point belongs to must also be interesting. We have thus arrived at a situation where we are now visualizing a whole time series, rather than a single point of data. Based on the amount of data conveyed in a time series, we

can relatively safely assume this time series to be better presented as a graph than as a textual description.

For the generation of the locator maps, the most significant problem is the determination of a good reference location. Optimally, the reference location should be known to almost all readers of the article. In this work, we used municipalities as reference locations and their population counts as a proxy for how well they are known. It is, however, not a perfect proxy and fails to identify cases where low-population municipalities are well known due to, e.g. cultural reasons or by their status as 'provincial capitals' of sorts.

Finding a better proxy is non-trivial. One way to tap into the cultural significance of a location is presented by Kim et al. [12], who identify smaller-scale reference locations using the number of photos on Flickr that were taken within 1 mile of a potential landmark and tag it in the photo. Other options include using Google search result numbers, Wikipedia article lengths and similar metrics as proxies. It is, however, unclear at this point how well these metrics capture the 'well known' aspect of a location, especially that of a municipality rather than a smaller landmark.

Furthermore, our approach to always select the closest reference location might be suboptimal when the map has an aspect ratio significantly different from 1:1. For example, if the map was twice as high as it is wide, selecting a reference location that is 100 kilometers south of the target location would likely be better than selecting one that is 100 kilometers to the east of the target location, as the first would allow for an overall higher level of zoom. It is furthermore possible that different map projections might cause problems. For example, the simple Mercator projection is significantly warped near the poles, so that a distance of 100km on the equator is significantly different from the same distance near the pole when measured on the map. The solution to these problems would be to calculate distances on the map projection, rather than on a globe, and using weights that account for the possible aspect ratio of the map. However, this can get complicated if the user is free to resize the map on, for example, a web page.

5.2 How to Visualize

While the 'how' of visualization is an easy step for a locator map, it is more challenging for graphing time series in a proper way. The difficulty of this process also seeps into what we discussed concerning deciding what to visualize. Part of that decision-making process is evaluating whether some information or dataset is best expressed as a visualization or not.

In this work, we constructed our graph content selection process so that it can result in a reasonable selection of data for visualization along with the produced text. As a result, our systems – at least insofar as our context is concerned – seems to always produce a sensible result. Yet, this comes at some cost to the flexibility and ability to adapt to different datasets without more programming work. For example, while the decision to always use the three different time series in a graph (target municipality, the national average, and the average of similar municipalities) produced very good graphs in this context, it is unlikely to generalize to all other datasets. With some datasets, other graph formats might even be more ideal. It seems that, at least now, the classic trade-off between 'specialized tool' and 'jack of all trades, master of none' exists here as well.

We used the municipality population data as a proxy for estimating the similarity between municipalities, which we believe to be a reasonable starting point. Yet, this does not reflect further socio-economical aspects that a human would consider when deciding whether two municipalities are 'similar' or not. Thus, by taking advantage of further statistical data using a wide variety of factors we can significantly improve the similarity measure. Other statistical measures can include information on whether the municipality is rural or urban and what is the socio-economic structure of the population and so forth. As for Finland, most of these factors are easy to discover from the datasets provided by Statistics Finland and will be explored as future work.

At the same time, we do believe that our graphs are well suited for visualizing the data in the system. The three time series in the single graph allow the users to observe trends both nationwide and in similar municipalities in a more clear way. The generated graphs in our system further show a smaller tooltip graph. This tooltip graph helps the user to distinguish between seasonal trends, e.g. crimes that are more common during the summer than the winter, and longer trends such as whether some crime is becoming more or less common in general. These interactions allow the user to get a large amount of information from a single graph, without having to query the system for more information.

6 CONCLUSIONS

In this work, we proposed a data-driven visualization subsystem for automated generation of visualizations to accompany automatically generated news articles. The proposed subsystem can generate locator maps, graphs and captions language-independently, and to a large degree domain-independently.

We have further demonstrated the feasibility of the approach via a practical implementation. We extended an automated natural language generation system producing news on crime statistics with a subsystem like the one proposed above. The system automatically determines newsworthy aspects to visualize, thus providing the reader with additional information that could be difficult to convey in a textual format. The system also automatically produces a locator map, which gives the reader a geographic context to where the story is taking place by displaying both the location discussed in the story and a completely automatically determined reference location the user is assumed to know.

Furthermore, we have identified additional opportunities to take the architecture into an even more data-driven direction, especially insofar as the selection of *how* to graph data is considered, thus potentially leading to even more domain-independent graph generation for automatically generated text.

ACKNOWLEDGMENTS

This work has been supported by the European Union's Horizon 2020 research and innovation programme under grant 770299 (NewsEye). It has also been supported by Business Finland (previously Tekes, the Finnish Funding Agency for Innovation) and multiple Finnish media industry companies as part of the 'Immersive Automation' research project.

REFERENCES

[1] H Arthur Robinson. 1958. *Elements of cartography*. John Wiley And Sons, Inc; New York.

[2] Jean-Daniel Fekete and Catherine Plaisant. 2003. Interactive information visualization of a million items. In *The Craft of Information Visualization*. Elsevier, 279–286.

[3] Stephen Few and Perceptual Edge. 2007. Data visualization: past, present, and future. *IBM Cognos Innovation Center* (2007).

[4] Statistics Finland. 2018. Statistics on offences and coercive measures. http://pxnet2.stat.fi/PXWeb/pxweb/en/StatFin/StatFin__oik__rpk.

[5] Finlex. 2018. Rikoslaki 19.12.1889/39. https://www.finlex.fi/fi/laki/ajantasa/1889/18890039001. Online; accessed 21 June 2018.

[6] Michael Friendly and Daniel J Denis. 2001. Milestones in the history of thematic cartography, statistical graphics, and data visualization. *URL http://www. datavis. ca/milestones* 32 (2001), 13.

[7] Tong Gao, Jessica R Hullman, Eytan Adar, Brent Hecht, and Nicholas Diakopoulos. 2014. NewsViews: an automated pipeline for creating custom geovisualizations for news. In *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 3005–3014.

[8] Jeffrey L Griffin and Robert L Stevenson. 1994. The effectiveness of locator maps in increasing reader understanding of the geography of foreign news. *Journalism Quarterly* 71, 4 (1994), 937–946.

[9] JS Highcharts. 2012. AS Highsoft Solutions. *http://www. highcharts. com* (2012).

[10] Jessica Hullman, Nicholas Diakopoulos, and Eytan Adar. 2013. Contextifier: automatic generation of annotated stock visualizations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2707–2716.

[11] Daniel A Keim. 2002. Information visualization and visual data mining. *IEEE Transactions on Visualization & Computer Graphics* 1 (2002), 1–8.

[12] Yea-Seul Kim, Jessica Hullman, and Maneesh Agrawala. 2016. Generating personalized spatial analogies for distances and areas. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 38–48.

[13] Leo Leppänen, Myriam Munezero, Mark Granroth-Wilding, and Hannu Toivonen. 2017. Data-Driven News Generation for Automated Journalism. In *Proceedings of the 10th International Conference on Natural Language Generation*. 188–197.

[14] Leo Leppänen, Myriam Munezero, Stefanie Sirén-Heikel, Mark Granroth-Wilding, and Hannu Toivonen. 2017. Finding and expressing news from structured data. In *Proceedings of the 21st International Academic Mindtrek Conference*. ACM, 174–183.

[15] Vibhu O Mittal, Giuseppe Carenini, Johanna D Moore, and Steven Roth. 1998. Describing complex charts in natural language: A caption generation system. *Computational Linguistics* 24, 3 (1998), 431–467.

[16] Matteo Picozzi, Nervo Verdezoto, Matti Pouke, Jarkko Vatjus-Anttila, and Aaron John Quigley. 2013. Traffic visualization-applying information visualization techniques to enhance traffic planning. In *GRAPP 2013 IVAPP 2013-Proceedings of the International Conference on Computer Graphics Theory and Applications and International Conference on Information Visualization Theory and Applications*. SciTePress.

[17] François Portet, Ehud Reiter, Albert Gatt, Jim Hunter, Somayajulu Sripada, Yvonne Freer, and Cindy Sykes. 2009. Automatic generation of textual summaries from neonatal intensive care data. *Artificial Intelligence* 173, 7-8 (2009), 789–816.

[18] Ehud Reiter and Robert Dale. 2000. *Building natural language generation systems*. Cambridge university press.

[19] C Carl Robusto. 1957. The cosine-haversine formula. *The American Mathematical Monthly* 64, 1 (1957), 38–40.

[20] Steven F Roth, John Kolojejchick, Joe Mattis, and Jade Goldstein. 1994. Interactive graphic design using automatic presentation knowledge. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 112–117.

[21] Mike Samuels and Nancy Samuels. 1975. *Seeing with the mind's eye: The history, techniques, and uses of visualization*. Random House Incorporated.

[22] Yiwen Sun, Jason Leigh, Andrew Johnson, and Sangyoon Lee. 2010. Articulate: A semi-automated model for translating natural language queries into meaningful visualizations. In *International Symposium on Smart Graphics*. Springer, 184–195.